|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project:** | | | | | Project Name | | | | | | | | | | | |
| **Customer:** | | | | | Customer Name | | | | | **Review**  **date:** | | | | yyyy-mm-dd | | |
|  | | | | | | | | | | | | | | | | |
| **Object to be reviewed:** | | | | | | | | | | | | | | | | |
| Document name: | | | Project\_Module | | | | | | | | | | | | | |
| Version: | | | 1.0 | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| **Participants:** | | | | | | | **Name** | | | **Org. unit** | | | | | **Review manager**  **(x)** | |
| Person responsible (author): | | | | | | | Gil-Dong, Hong | | | R&D | | | | | x | |
| SW Project Manager: | | | | | | |  | | |  | | | | |  | |
| SW Analyst: | | | | | | |  | | |  | | | | |  | |
| SW Integrator: | | | | | | |  | | |  | | | | |  | |
| Other reviewer(s): | | | | | | |  | | |  | | | | |  | |
|  | | | | | | | | | | | | | | | | |
| **Review type:** | | | | | | | | | | | | | **Project status:** | | | |
|  | | | |  | |  | |  |  | |  | | Phase: | | |  |
|  | | | | | | | | | | | | | | | | |
| **Last review of the object to be reviewed:** | | | | | | | | | | | | | | | | |
| Performed on: | |  | | | | | | | | | | | | | | |
| Doc. name: | |  | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| **Follow Up:** | | | | | | | | | | | | | | | | |
|  |  | | | | | | | | | | | Date: | | |  | |
|  |  | | | | | | | | | | | Date: | | |  | |

| **Checklist 'Review Meeting'** | **OK?** | | **Action/comments** |
| --- | --- | --- | --- |
| **yes** | **no** |
| Have the coding rules according to the coding standard (C style guide) been met (structure, instructions, naming)? |  |  | If this checklist is not applicable, you should leave comment and 'NA'.  If you have comment then you should leave comment at this. |
| Has a static code analysis (MISRA with QA-C) been performed on the relevant C code? This includes checking functional size and controlling warnings!. |  |  |  |
| Have all errors/warnings been checked and removed, which occurred during the static code analysis? |  |  |  |
| Have all necessary deviations (messages and code metrics) found out by static code analysis been reviewed and removed if possible? |  |  |  |
| Have all deviations which have been reviewed and can’t been removed been documented with a reason? |  |  |  |
| Have all naming convention been applied? |  |  |  |
| Is the implementation of the module/the functions consistent with the design? |  |  |  |
| Are the interrelations of code and design consistent? |  |  |  |
| Do metrics exist for the code and are the appropriate limits observed? |  |  |  |
| Have all interface restrictions been applied?! |  |  |  |
| Are excessively deep or infinite recursion been used? |  |  |  |
| Is the documentation guideline applied? |  |  |  |
| Has function overlapping been avoided in the coding of the modules? |  |  |  |
| Are all constants and variables consistent with the data dictionary? |  |  |  |
| For existing variables and memory has always the standard call interfaces been used? |  |  |  |
| Is module independence in accordance with the design kept? |  |  |  |
| Are all algorithms correctly implemented? |  |  |  |
| Are possible overflows or range exceeding intercepted? |  |  |  |
| Have risks been detected and assessed? |  |  |  |
| Does no recurring or redundant code exist? |  |  |  |
| Are macros, procedures, functions and other reusable code components used to prevent recurring and redundant code? |  |  |  |
| Are sections of code 'commented out'?. |  |  |  |
| Does in each software function only exist one entry or exit point and other exit point are exclusively used to deal with errors or exceptional situations? |  |  |  |
| Is control always returned to the calling instance after execution of a sub-function? |  |  |  |
| Has it been ensured that no port is accessed outside the driver layer? |  |  |  |
| Do all program parts which are waiting for a result have timeouts and is a defined abortion planned in case of error? |  |  |  |
| Is the accuracy of calculations (e.g. floating point types) sufficient? |  |  |  |
| Are all variables initialized, before they are used? |  |  |  |
| Are non-defined input values intercepted? |  |  |  |
| If the bitwise operators ~ and << are applied to an operand of underlying type unsigned char or unsigned short is the result cast to the underlying type of the operand? |  |  |  |
| Is pointer arithmetic only applied to pointers that address an array or array element? |  |  |  |
| Is pointer subtraction only applied to pointers that address elements of the same array? |  |  |  |
| Are “>, >=, <, <=” not applied to pointer types except where they point to the same array? |  |  |  |
| Are there any interrupts witch could endanger the completion of a task and consistency of data?!  Is data consistency (information to information, information to control data, e.g events) always guaranteed in all closed function blocks (e.g. tasks)? |  |  |  |
| Have all available bug lists, Errata Sheets and change notices for the Microcontroller been known, analysed and considered? |  |  |  |
| Does an appropriate workaround or patch exist for all compiler bugs which affect the resulting executable? |  |  |  |
| If a workaround or patch has to be used, is this documented? |  |  |  |
| PS for review: added: 2.4, 3.6, 5.7, 10.5, 17.1, 17.2, 17.3, 18.2, 18.3 |  |  |  |

|  | **Complaint / fault** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Page/**  **Line** | **Issue** | **Safety**  **relevant** | | **Action** | | | |
| **yes** | **no** | **Solution** | **Responsible person** | **Completion deadline** | **Done**  **(🗸)** |
| 1 | 15/20 | There is no diagram for explian of module. |  | x | Add diagram for module. | Gil-Dong, Hong | yyyy-mm-dd | **🗸** |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |